

Structural Return Maximization for Reinforcement Learning

Joshua Joseph, Javier Velez, Nicholas Roy

Massachusetts Institute of Technology
{jmjoseph, velezj, nickroy}@mit.edu

May 13, 2014

Abstract

Batch Reinforcement Learning (RL) algorithms attempt to choose a policy from a designer-provided class of policies given a fixed set of training data. Choosing the policy which maximizes an estimate of return often leads to over-fitting when only limited data is available, due to the size of the policy class in relation to the amount of data available. In this work, we focus on learning policy classes that are appropriately sized to the amount of data available. We accomplish this by using the principle of Structural Risk Minimization, from Statistical Learning Theory, which uses Rademacher complexity to identify a policy class that maximizes a bound on the return of the best policy in the chosen policy class, given the available data. Unlike similar batch RL approaches, our bound on return requires only extremely weak assumptions on the true system.

1 Introduction

Reinforcement Learning (RL) (Sutton & Barto, 1998) is a framework for sequential decision making under uncertainty with the objective of finding a policy that maximizes the sum of rewards, or *return*, of an agent. A straightforward model-based approach to batch RL, where the algorithm learns a policy from a fixed set of data, is to fit a dynamics model by minimizing a form of prediction error (*e.g.*, minimum squared error) and then compute the optimal policy with respect to the learned model (Bertsekas, 2000). As discussed in Baxter & Bartlett (2001) and Joseph et al. (2013), learning a model for RL by minimizing prediction error can result in a policy that performs arbitrarily poorly for unfavorably chosen model classes. To overcome this limitation, a second approach is to not use a model and directly learn the policy from a policy class that explicitly maximizes an estimate of return (Meuleau et al., 2000).

With limited data, approaches that explicitly maximize estimated return are vulnerable to learning policies which perform poorly since the return cannot be confidently estimated. We overcome this problem by applying the principle of Structural Risk Minimization (SRM) (Vapnik, 1998), which, in terms of RL, states that instead of choosing

the policy which maximizes the estimated return we should instead maximize a bound on return. In SRM the policy class size is treated as a controlling variable in the optimization of the bound, allowing us to naturally trade-off between estimated performance and estimation confidence. By controlling policy class size in this principled way we can overcome the poor performance of approaches which explicitly maximize estimated return with small amounts of data.

The main contribution of this work is a batch RL algorithm which has bounded true return under extremely weak assumptions, unlike standard batch RL approaches. Our algorithm is the result of applying the principle of SRM to RL, which previously has only been studied in the context of classification and regression. We first show a bound on the return of a single policy from a fixed policy class based on a technique called Model-Free Monte Carlo (Fonteneau et al., 2012). We then map RL to classification, allowing us to transfer generalization bounds based on Rademacher complexity (Bartlett & Mendelson, 2003) which results in a bound on the return of any policy from a policy class. Given a structure of policy classes, we then apply the principle of SRM to find the highest performing policy from the family of policy classes.

Section 2 reviews Structural Risk Minimization in the context of classification. We move to RL in Section 3 and show the bound on the return of a policy. Section 4 ties together the previous two sections to provide a bound on return for a policy from a structure of policy classes and discusses some of the natural policy class structures that exist in RL. Section 5 first demonstrates our approach on a simple domain to build intuition for the reader and then validates its performance on increasingly difficult problems. Section 6 discusses related work and Section 7 concludes the paper.

2 Structural Risk Minimization for Classifier Learning

In this section we review Structural Risk Minimization for classification for completeness and to ensure that the parallels presented in Section 4 are clear for the reader. Classification is the problem of deciding on an output, $y \in \mathcal{Y}$, for a given input, $x \in \mathcal{X}$. The performance of a decision rule $f : \mathcal{X} \rightarrow \mathcal{Y}$ is measured using risk, \mathcal{R} , defined as

$$\mathcal{R}(f) = \int \mathcal{L}(y, f(x))p(x, y)dx dy \quad (1)$$

where $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is the loss function and $p(x, y)$ is the data generating distribution. For a class of decision rules, \mathcal{F} , the objective of classification is to select the decision rule which minimizes risk or, more formally,

$$f^* = \arg \min_{f \in \mathcal{F}} \mathcal{R}(f). \quad (2)$$

2.1 Empirical Risk Minimization

Commonly, the distribution $p(x, y)$ in Equation 1 is unknown, and we therefore are unable solve Equation 2 using Equation 1. Given a dataset $\mathcal{D} = \{(x^1, y^1), (x^2, y^2), \dots, (x^N, y^N)\}$

where (x^n, y^n) is drawn i.i.d. from $p(x, y)$, Equation 1 can be approximated by empirical risk

$$R_{emp}(f; \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y^n, f(x^n)). \quad (3)$$

By using empirical risk as an estimate of risk we can attempt to solve Equation 2 using the principle of Empirical Risk Minimization (ERM) (Vapnik, 1998) where

$$\hat{f} = \arg \min_{f \in \mathcal{F}} R_{emp}(f; \mathcal{D}). \quad (4)$$

In Section 3 we see that there is a clear analogy between this result and how a policy's return is estimated in RL.

2.2 Bounding the Risk of a Classifier

We can bound risk (Equation 1) using using empirical risk (Equation 3) with a straightforward application of Hoeffding's inequality

$$\mathcal{R}(f) \leq R_{emp}(f; \mathcal{D}) + \sqrt{\frac{-\ln \delta}{2N}} \quad (5)$$

which holds with probability $1 - \delta$. Since Equation 4 is used to choose $f \in \mathcal{F}$, we need to ensure that $\mathcal{R}(f)$ is bounded for *all* $f \in \mathcal{F}$ (not just for a single f as Equation 5 guarantees). Bounds of this form ($\forall f \in \mathcal{F}$) can be written as

$$\mathcal{R}(f) \leq R_{emp}(f; \mathcal{D}) + \Omega(\mathcal{L} \circ \mathcal{F}, \mathcal{D}, \delta) \quad (6)$$

where Ω can be thought of as a complexity penalty on the size of $\mathcal{L} \circ \mathcal{F} = \{\mathcal{L}(\cdot, f(\cdot)) : f \in \mathcal{F}\}$ and the bound holds with probability $1 - \delta$.

Section 2.2.1 and 2.2.2 describes a specific forms of Ω using Vapnik-Chervonenkis Dimension and Rademacher complexity, which we chose due to their popularity in the literature although many additional bounds are known, *e.g.*, maximum discrepancy (Bartlett et al., 2002a), local Rademacher complexity (Bartlett et al., 2002b), Gaussian complexity (Bartlett & Mendelson, 2003).

2.2.1 Vapnik-Chervonenkis Dimension

A well studied bound from Vapnik (1995) that takes the form of Equation 6 uses

$$\Omega(\mathcal{L} \circ \mathcal{F}, \mathcal{D}, \delta) = \frac{B - A}{2} \sqrt{4 \frac{h \left(\ln \left(\frac{2N}{h} \right) + 1 \right) - \ln(\delta/4)}{N}} \quad (7)$$

where $A \leq \mathcal{L} \circ \mathcal{F} \leq B$ and h is the Vapnik-Chervonenkis (VC) dimension of $\mathcal{L} \circ \mathcal{F}$ (see Vapnik (1998) for a thorough description of VC dimension).

2.2.2 Rademacher Complexity

In contrast to the well studied bound from Vapnik (1995) which depends on the Vapnik-Chervonenkis (VC) dimension of $\mathcal{L} \circ \mathcal{F}$, Bartlett & Mendelson (2003) provide a bound based on the Rademacher complexity of $\mathcal{L} \circ \mathcal{F}$, a quantity that can be straightforwardly estimated from the dataset. Their bound, which takes the form of Equation 6, uses

$$\Omega(\mathcal{L} \circ \mathcal{F}, \mathcal{D}, \delta) = R^N(\mathcal{L} \circ \mathcal{F}) + \sqrt{\frac{-8 \ln(2\delta)}{N}} \quad (8)$$

where $0 \leq \mathcal{L} \circ \mathcal{F} \leq 1$, and σ_n is a uniform random variable over $\{-1, +1\}$. $R^N(\mathcal{F})$, the Rademacher complexity of \mathcal{F} , can be estimated using

$$\begin{aligned} R^N(\mathcal{F}) &= E_{\sigma_{1:N}, x^{1:N}} \left[\sup_{f \in \mathcal{F}} \frac{2}{N} \left| \sum_{n=1}^N \sigma_n f(x^n) \right| \right], \\ \hat{R}^N(\mathcal{F}; \mathcal{D}) &= E_{\sigma_{1:N}} \left[\sup_{f \in \mathcal{F}} \frac{2}{N} \left| \sum_{n=1}^N \sigma_n f(x^n) \right| \middle| x^{1:N} \right]. \end{aligned} \quad (9)$$

Bartlett & Mendelson (2003) also show that the error from estimating Rademacher complexity using the right hand side of Equation 9 is bounded with probability $1 - \delta$ by

$$R^N(\mathcal{F}) \leq \hat{R}^N(\mathcal{F}; \mathcal{D}) + \sqrt{\frac{-8 \ln \delta}{N}}. \quad (10)$$

2.3 Structural Risk Minimization

As discussed in Vapnik (1995), the principle of ERM is only intended to be used with a large amount of data (relative to the size of \mathcal{F}). With a small data set, a small value of $R_{emp}(f; \mathcal{D})$ does not guarantee that $\mathcal{R}(f)$ will be small and therefore solving Equation 4 says little about the generalization of f . The principle of Structural Risk Minimization (SRM) states that since we cannot guarantee the generalization of ERM under limited data we should explicitly minimize the bound on generalization (Equation 6) by using a *structure* of function classes.

A structure of function classes is defined as a collection of nested subsets of functions $S_1 \subseteq S_2 \subseteq \dots \subseteq S_k \subseteq \dots$ where $S_k = \mathcal{L} \circ \mathcal{F}_k$. For example, a structure of radial basis functions created by placing increasing limits on the magnitude of the basis functions. SRM then treats the capacity of $\mathcal{L} \circ \mathcal{F}_k$ as a controlling variable and minimizes Equation 6 for each S_k such that

$$\begin{aligned} \hat{k} &= \arg \min_k R_{emp}(\hat{f}_k; \mathcal{D}) + \Omega(\mathcal{L} \circ \mathcal{F}_k, \mathcal{D}, \delta) \\ \hat{f}_k &= \arg \min_{f \in \mathcal{F}_k} R_{emp}(f; \mathcal{D}). \end{aligned} \quad (11)$$

To solve Equation 11 we must solve both equations jointly. One can imagine enumerating k , finding \hat{f}_k for each k , and choosing the corresponding \hat{f}_k which minimizes $R_{emp}(\hat{f}_k; \mathcal{D}) + \Omega(\mathcal{L} \circ \mathcal{F}_k, \mathcal{D}, \delta)$.

3 Bounding the Return of a Policy

Section 2 allowed us to bound classification performance given small amounts of data; we now turn our attention to bounding policy performance. A finite time Markov Decision Process (MDP) is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{W}, m, \rho, s_{start}, T)$ where \mathcal{S} is the state space, \mathcal{A} is the action space, \mathcal{W} is the disturbance space¹, $m : \mathcal{S} \times \mathcal{A} \times \mathcal{W} \rightarrow \mathcal{S}$ is the dynamics model, $\rho : \mathcal{S} \rightarrow \mathbb{R}$ is the reward function, $s_{start} \in \mathcal{S}$ is the starting state, and T is the maximum length of an episode². For a policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, we define its return³ as

$$V(\pi) = \int_{w_{0:T-1}} \left[\rho(s_0) + \sum_{t=0}^{T-1} \rho(m(s_t, \pi(s_t), w_t)) \right] p(w_{0:T-1}) dw_{0:T-1} \quad (12)$$

$$= E_{w_{0:T-1}} \left[\sum_{t=0}^{T-1} \rho(s_t) \middle| s_0 = s_{start}, s_{t+1} = m(s_t, \pi(s_t), w_t) \right]. \quad (13)$$

We call the sequence $e_\pi = \{s_0, a_0, s_1, a_1, \dots, s_{T-1}, a_{T-1}\}$ an episode of data where $s_{t+1} = m(s_t, a_t, w_t)$ and $a_t = \pi(s_t)$.

Throughout this work we assume that the return is bounded by $A \leq \sum_{t=0}^{T-1} \rho(s_t) \leq B$ and that the dynamics model, reward function, and policy are Lipschitz continuous with constants L_m, L_ρ , and L_π , respectively (Fonteneau et al., 2012), where

$$\begin{aligned} \|m(s, a, w) - m(s', a', w)\|_{\mathcal{S}} &\leq L_m(\|s - s'\|_{\mathcal{S}} + \|a - a'\|_{\mathcal{A}}), \\ |\rho(s, a, w) - \rho(s', a', w)| &\leq L_\rho(\|s - s'\|_{\mathcal{S}} + \|a - a'\|_{\mathcal{A}}), \\ \|\pi(t, s) - \pi(t, s')\|_{\mathcal{A}} &\leq L_\pi\|s - s'\|_{\mathcal{S}}, \\ \forall (s, s', a, a', w) &\in \mathcal{S}^2 \times \mathcal{A}^2 \times \mathcal{W}, \forall t \in \{0, \dots, T-1\}. \end{aligned}$$

The objective of an MDP is to find the policy

$$\pi^* = \arg \max_{\pi \in \Pi} V(\pi) \quad (14)$$

from a given policy class, Π . Typically in RL the dynamics model, m , is unavailable to us and therefore we are unable to use Equation 13 to solve Equation 14.

¹The disturbance space is introduced so we may assume the dynamics model is deterministic and add noise through w . This is primarily done to facilitate theoretical analysis and is equivalent to the standard RL notation which uses a stochastic dynamics model that does not include w .

²For simplicity we assume ρ and $s_0 = s_{start}$ are known and deterministic and ρ is only a function of the current state. Without loss of generality this allows us to write V as a scalar everywhere. This work can be straightforwardly extended to an unknown and stochastic ρ and s_{start} and the more general $\rho : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$.

³ $V(\pi)$ is shorthand for $V^\pi(s_{start})$, the expected sum of rewards from following policy π from state s_{start} .

3.1 Estimating the Return of a Policy

To overcome not knowing m in Equation 13 we commonly use data of interactions with m to estimate $V(\pi)$, called policy evaluation, and then solve Equation 14. A difficulty in estimating $V(\pi)$ lies in how the data was collected, or, more precisely, which policy was used to generate the data. We discuss two different types of policy evaluation, *on-policy*, where the data used to estimate $V(\pi)$ is generated using π , and *off-policy*, where the data is collected using any policy.

3.1.1 On-Policy Policy Evaluation

Given a set of N episodes of data $\{e_\pi^1, e_\pi^2, \dots, e_\pi^N\}$ collected using π , we can estimate $V(\pi)$ using empirical return (analogous to Equation 3) where

$$V_{emp}(\pi; \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N \sum_{t=0}^{T-1} \rho(s_t^n), \quad (15)$$

$$V(\pi) \geq V_{emp}(\pi; \mathcal{D}) - (B - A) \sqrt{\frac{-\ln \delta}{2N}}, \quad (16)$$

$e_\pi^n = \{s_0^n, a_0^n, \dots, s_{T-1}^n, a_{T-1}^n\}$, $s_{t+1}^n = m(s_t^n, a_t^n, w_t^n)$, $a_t^n = \pi(s_t^n)$, and Equation 16 holds with probability $1 - \delta$ by Hoeffding's inequality (Hoeffding, 1963). This approach, where episodes are generated on-policy and then the return is estimated using Equation 15, is called Monte Carlo policy evaluation (Sutton & Barto, 1998). Since we do not make any assumptions about the policies that will be evaluated nor the policy under which the data was generated, we cannot directly use Equations 15 and 16 but we will build upon them in the following sections.

3.1.2 Off-Policy Policy Evaluation

Naively, using Equation 15 to approximate and solve Equation 14, would require N data for each $\pi \in \Pi$, an infinite amount of data for infinite policy classes. Off-policy policy evaluation aims to alleviate this issue by estimating $V(\pi)$ using episodes $e_{\pi_1}^1, e_{\pi_2}^2, \dots, e_{\pi_N}^N$ where π_1, \dots, π_N may be different than π . To perform off-policy evaluation, we use an approach called Model-Free Monte Carlo-like policy evaluation (MFMC) (Fonteneau et al., 2012), which attempts to approximate the estimator from Equation 15 by piecing together artificial episodes of on-policy data from off-policy, batch data.

Consider a set of data $\{e_{\pi_1}^1, e_{\pi_2}^2, \dots, e_{\pi_N}^N\}$, which we re-index as one-step transitions

$$\mathcal{D} = \{(s_0, a_0, s_1), \dots, (s_{NT-1}, a_{NT-1}, s_{NT})\}.$$

To evaluate a policy, π , MFMC uses a distance function

$$\Delta((s, a), (s', a')) = \|s - s'\|_{\mathcal{S}} + \|a - a'\|_{\mathcal{A}}$$

and pieces together \tilde{N} artificial episodes from \mathcal{D} such that $\tilde{e}_\pi^n = \{\tilde{s}_0^n, \tilde{a}_0^n, \dots, \tilde{s}_T^n, \tilde{a}_T^n\}$ is an artificial on-policy episode approximating an episode π for $n = 1, \dots, \tilde{N}$. To

construct \tilde{e}_π^n , MFMC starts with $\tilde{s}_0^n = s_{start}$ and for $t = 1, \dots, T$ we find

$$\tilde{s}_{t+1}^n = \arg \min_{s': (s, a, s') \in \mathcal{D}} \Delta((s, a), (\tilde{s}_t^n, \tilde{a}_t^n)) \quad (17)$$

where $\tilde{a}_t^n = \pi(\tilde{s}_t^n)$ and once a transition, (s, a, s') , is chosen using Equation 17 it is removed from \mathcal{D} . Following the construction of episodes $\tilde{e}_\pi^1, \dots, \tilde{e}_\pi^{\tilde{N}}$, MFMC estimates the return of π using

$$V_{MFMC}(\pi; \mathcal{D}) = \frac{1}{\tilde{N}} \sum_{n=1}^{\tilde{N}} \sum_{t=0}^{T-1} \rho(\tilde{s}_t^n). \quad (18)$$

We can bound the return using Theorem 4.1, Lemma A.1, and Lemma A.2 of Fonteneau et al. (2010) and say that

$$V(\pi) \geq E_{\mathcal{D}} [V_{MFMC}(\pi)] - d(\pi; \mathcal{D}) \quad (19)$$

where

$$d(\pi; \mathcal{D}) = \max_{n \leq \tilde{N}} \sum_{t=0}^{T-1} L_{T-t} \delta_t^n, \\ \delta_t^n = \min_{s': (s, a, s') \in \mathcal{D}} \Delta((s, a), (\tilde{s}_t^n, \tilde{a}_t^n))$$

for each \tilde{s}_{t+1}^n chosen using Equation 17, and

$$L_{T-t} = L_\rho \sum_{i=0}^{T-t-1} [L_m(1 + L_\pi)]^i.$$

The term $d(\pi; \mathcal{D})$ is the maximum deviation between the true return of any policy and the expected MFMC estimate of return of that policy. See Fonteneau et al. (2010, 2012) for a discussion regarding the choice of Δ and \tilde{N} .

3.1.3 Probabilistic Bound of the MFMC Estimator

Unfortunately, the bound provided in Equation 19 only allows us to bound the return using the *expectation* of the MFMC estimate, not the *realized* estimate based on the data, which is required in Section 4. We present such a bound, beginning with Hoeffding's inequality (Hoeffding, 1963)⁴,

$$e^{2N(\epsilon - d(\pi; \mathcal{D}))^2 / (B-A)^2} \\ > \Pr [V_{MFMC}(\pi; \mathcal{D}) - E[V_{MFMC}(\pi)] + d(\pi; \mathcal{D}) > \epsilon] \quad (20)$$

$$\geq \Pr [V_{MFMC}(\pi; \mathcal{D}) - V(\pi) > \epsilon] \quad (21)$$

⁴Note MFMC only allows a single transition to be used once in constructing a set of episodes. Therefore, the return from the episodes do not violate the independence assumption required to use Hoeffding's inequality.

where we move from Equation 20 to Equation 21 using Equation 19. Setting $\delta = e^{2N(\epsilon - d(\pi; \mathcal{D}))^2 / (B-A)^2}$, solving for ϵ , and substituting the quantity into Equation 21 we see that with at least probability $1 - \delta$

$$V(\pi) \geq V_{MFMC}(\pi; \mathcal{D}) - d(\pi; \mathcal{D}) - (B - A) \sqrt{\frac{-\ln \delta}{2N}}. \quad (22)$$

While Equation 22 is useful for bounding the return estimate using MFMC, in Section 4 we will require a bound between $V_{MFMC}(\pi; \mathcal{D})$ and $V_{emp}(\pi)$. By combining Equations 16 and 22, we have with probability $1 - 2\delta$

$$\begin{aligned} V_{emp}(\pi; \mathcal{D}) &\geq V_{MFMC}(\pi; \mathcal{D}) - d(\pi; \mathcal{D}) \\ &\quad - 2(B - A) \sqrt{\frac{-\ln \delta}{2N}}. \end{aligned} \quad (23)$$

4 Structural Return Maximization for Policy Learning

Sections 2 and 3 provide intuition for the similarities between classification and RL, *e.g.*, in classification we choose $f \in \mathcal{F}$ using Equations 1 and 2, and in RL we choose $\pi \in \Pi$ using Equations 13 and 14. In this section we aim to formalize the relationship between classification and RL and by doing so we are able to use Structural Risk Minimization (Section 2.3) to learn policies drawn from policy classes that are appropriately sized given the available data.

4.1 Mapping Classification to Reinforcement Learning

The difficulty in mapping classification to RL is most easily seen when we consider the bounds in Sections 2.2.1 and 2.2.2, for which we need to know the RL equivalent of $\mathcal{L} \circ \mathcal{F}$ to either compute its VC dimension (for Equation 7) or its Rademacher complexity (for Equation 8). To show the mapping we begin by defining the *return function*,

$$G(s_0, w_{0:T-1}, a_{0:T-1}) \triangleq \rho(s_0) + \sum_{t=0}^{T-1} \rho(m(s_t, a_t, w_t)), \quad (24)$$

and show that the classification objective of minimizing risk is equivalent to the RL objective of maximizing return. Using Equation 2 and a loss function that does not depend on y ⁵, we set $\mathcal{L}(y, f(x)) = \tilde{\mathcal{L}}(x, f)$ and see that

$$\mathcal{R}(f) = \int \tilde{\mathcal{L}}(x, f) p(x) dx \quad (25)$$

$$= \int -\tilde{G}(s_0, w_{0:T-1}, f) p(s_0, w_{0:T-1}) dw_{0:T-1} \quad (26)$$

$$= -V(f) \quad (27)$$

⁵While it may seem that writing \mathcal{L} without y is an abuse of notation, if instead we view \mathcal{L} as a measure of performance the relationship between RL and classification becomes more clear.

where we go from Equation 25 to Equation 26 by setting $x = [s_0, w_{0:T-1}]$ and noting that $\tilde{\mathcal{L}}(x, f) = -\tilde{G}(s_0, w_{0:T-1}, f) = -G(s_0, w_{0:T-1}, a_{0:T-1})$ for $a_t = f(s_t)$, and we move from Equation 26 to Equation 27 using Equation 13. Therefore, minimizing $\mathcal{R}(f)$ for $f \in \mathcal{F}$ is identical to maximizing $V(f)$ for $f \in \mathcal{F}$. We see that G is the term in brackets in Equation 12; G encodes both the reward function and dynamics model and is analogous to classification's \mathcal{L} . This is a crucial relationship since transferring the bounds from Section 2.2 depends on our being able to calculate the RL equivalent of $\mathcal{L} \circ \mathcal{F}$, which we denote $G \circ \Pi$ ⁶.

4.2 Bound on Return for all Policies in a Policy Class

Using the mapping from the previous section we can rewrite Equations 3 and 6 for RL as

$$\begin{aligned} V_{emp}(\pi; \mathcal{D}) &= \frac{1}{N} \sum_{n=1}^N G(s_0, w_{0:T-1}^{\pi, n}, a_{0:T-1}^{\pi, n}) \\ &= \frac{1}{N} \sum_{n=1}^N \sum_{t=0}^T \rho(s_t^{\pi, n}) \end{aligned} \quad (28)$$

$$V(\pi) \geq V_{emp}(\pi; \mathcal{D}) - \Omega(G \circ \Pi, \mathcal{D}, \delta). \quad (29)$$

where $s_{t+1}^{\pi, n} = m((s_t^{\pi, n}, \pi(s_t^{\pi, n}), w_{0:T-1}^n)$. Note that in Equation 28 $w_{0:T-1}^n$ is needed to compute the empirical return which typically is not observed in practice. Since we assume the state is fully observable, we can use Equation 28 to calculate empirical return. Combining Equations 23 and 29,

$$\begin{aligned} V(\pi) &\geq V_{MFC}(\pi; \mathcal{D}) - d(\pi; \mathcal{D}) \\ &\quad - 2(B - A) \sqrt{\frac{-\ln \delta}{2N}} - \Omega(G \circ \Pi, \mathcal{D}, \delta) \end{aligned} \quad (30)$$

which holds with probability $1 - 3\delta$ and is the bound on return for all policies in Π . Section 4.4 describes the use Rademacher complexity (Section 2.2.2) to compute Ω in Equation 30.

4.3 Bound Based on VC Dimension for RL

To use the bound described in Section 2.2.1 we need to know the VC dimension of $G \circ \Pi$. Unfortunately, the VC dimension is only known for specific function classes (*e.g.*, linear indicator functions (Vapnik, 1998)), and, since the only assumptions we made about functional form of ρ , m , and π is that they are Lipschitz continuous, $G \circ \Pi$ will not in general have known VC dimension.

There also exist known bounds on the VC dimension for *e.g.*, neural networks (Anthony & Bartlett, 1999), decision trees (Asian et al., 2009), support vector machines

⁶Note that it is difficult to tell how deep the connection between classification and RL is and, while we think this will prove an interesting line of future research, the purpose of this connection is solely to provide us with the machinery necessary for the bounds.

(Vapnik, 1998), smoothly parameterized function classes (Lee et al., 1995), but, again, due to our relatively few assumptions on ρ , m , and π , $G \circ \Pi$ is not a function class with a known bound on the VC dimension. Cherkassky & Mulier (1998) notes that for some classification problems, the VC dimension of \mathcal{F} can be used as a reasonable approximation for the VC dimension $\mathcal{L} \circ \mathcal{F}$, but we have no evidence to support this being an accurate approximation when used in RL.

Other work has been done to estimate the VC dimension from data (Shao et al., 1969; Vapnik et al., 1994) and bound the VC dimension estimate (McDonald et al., 2011). While, in principle, we are able to estimate the VC dimension using one of these techniques, the approach described in Section 4.4 is a far simpler method for computing Ω in Equation 29 based on data.

4.4 Bound Based on Rademacher Complexity for RL

Using the Rademacher complexity bound (Section 2.2.2) allows us to calculate Ω (Equation 29) based on data. The only remaining piece is how to calculate the summation inside the absolute value sign of Equation 9 for RL. Mapping the Rademacher complexity estimator (Equation 9) into RL yields

$$\hat{R}^N(G \circ \Pi; \mathcal{D}) = E_{\sigma_{1:\tilde{N}}} \left[\sup_{\pi \in \Pi} \frac{2}{\tilde{N}} \left| \sum_{n=1}^{\tilde{N}} \sigma_n \sum_{t=0}^{T-1} \rho(s_t^n) \right| \middle| \mathcal{D} \right]. \quad (31)$$

Therefore, with probability $1 - \delta$

$$\begin{aligned} \hat{R}^N(G \circ \Pi; \mathcal{D}) &\geq \\ &- 2 + E_{\sigma_{1:\tilde{N}}} \left[\sup_{\pi \in \Pi} \frac{2}{\tilde{N}} \left| \sum_{n=1}^{\tilde{N}} \sigma_n \left(V_{MFMC}(\pi; \mathcal{D}) \right. \right. \right. \\ &\quad \left. \left. \left. - d(\pi; \mathcal{D}) - 2\sqrt{\frac{-\ln \delta}{2N}} \right) \right| \middle| \mathcal{D} \right] \end{aligned} \quad (32)$$

where we move from Equation 31 to Equation 32 using

$$\begin{aligned} \sigma_n \sum_{t=0}^{T-1} \rho(s_t^n) &\geq -\tilde{N} + \sigma_n \left(V_{MFMC}(\pi; \mathcal{D}) \right. \\ &\quad \left. - d(\pi; \mathcal{D}) - 2\sqrt{\frac{-\ln \delta}{2N}} \right) \end{aligned} \quad (33)$$

from Equation 23 $\forall n \in \{1, \dots, \tilde{N}\}$ and we assumed $-1 \leq \sum_{t=0}^{T-1} \rho(s_t) \leq 0$ for simplicity.

4.5 Structures of Policy Classes

In Section 2.3 we defined a structure of function classes, from which a function class and function from that class are chosen using Equation 11. To use structural risk mini-

mization for RL, we must similarly define a structure,

$$\Pi_1 \subseteq \Pi_2 \subseteq \dots \subset \Pi_k \subseteq \dots, \quad (34)$$

of policy classes⁷. For RL we add the additional constraint that the policies must be Lipschitz continuous (Section 3) in order to use the bound provided in Section 3.1.2. Note that the structure (*e.g.*, the indexing order $1, \dots, k, \dots$) must be specified *before* any data is seen.

Fortunately, some function classes have a “natural” ordering which may be taken advantage of, for example support vector machines (Vapnik, 1998) use decreasing margin size as an ordering of the structure. In RL, many common policy representations contain natural structure and are also Lipschitz continuous. Consider a policy class consisting of the linear combinations of radial basis functions (Menache et al., 2005). This class is Lipschitz continuous and using this representation, we may impose a structure by progressively increasing a limit on the magnitude of all basis functions, therefore high k allows for a greater range of actions a policy may choose. This policy class consists of policies of the form

$$\pi(s) = \sum_{i=1}^M \phi_i e^{-c(s-\bar{s}_i)^2} \quad (35)$$

where $c \geq 0$, $\bar{s}_i \in \mathcal{S}$ are fixed beforehand and the progressively increasing limit $l_k \geq |\phi_i|$, $i \in \{1, \dots, M\}$ and k is the index of the policy class structure.

A second policy representation which meets our requirements consists of policies of the form

$$\pi(s) = \sum_{i=1}^M \frac{\phi_i}{\|s - l_i\|_{\mathcal{S}}} \quad (36)$$

where $l_i \in \mathcal{S}$ are fixed beforehand and a policy of this representation is described by set $\phi_i \in \mathcal{A}$, $i \in \{1, \dots, M\}$. Using this representation, we then present two possible structures, the second of which we use for the experiments in Section 5. The first places a cap on ϕ_i , where $-a_k \leq \phi_i \leq a_k$, $\forall i \in \{1, \dots, M\}$ and $a_k \leq a_{k+1}$. The second structure “ties” together ϕ_i , $i \in \{1, \dots, M\}$ such that for $k = 1$, $\phi_1 = \phi_2 = \dots = \phi_M$. For $k = 2$, we untie ϕ_1 but still maintain that $\phi_2 = \dots = \phi_M$ and continue for the remaining k such that Equation 34 is maintained.

Even though the representations described in this section are natural structures that can be used for many RL policy classes, the performance of each will strongly depend on the problem. In general, choosing a policy class for a RL problem is often difficult and therefore it often must be left up to a designer. We leave further investigation into automatically constructing structures or making the structure data-dependent (*e.g.*, Shawe-Taylor et al. (1996)) to future work.

⁷Note that the structure is technically over $G \circ \Pi_k$ but $\Pi_i \subseteq \Pi_j \implies G \circ \Pi_i \subseteq G \circ \Pi_j$.

4.6 Finding the Best Policy from a Structure of Policy Classes

Using a structure of policy classes as described in the previous section, we may now reformulate the objective of SRM into Structural Return Maximization for RL as

$$\begin{aligned} \hat{k} = \arg \min_k V_{MFMC}(\hat{\pi}_k; \mathcal{D}) - d(\hat{\pi}_k; \mathcal{D}) \\ - 2(B - A) \sqrt{\frac{-\ln \delta}{2N}} - \Omega(G \circ \Pi_k, \mathcal{D}, \delta) \end{aligned} \quad (37)$$

$$\hat{\pi}_k = \arg \min_{\pi \in \Pi_k} V_{MFMC}(\pi; \mathcal{D}) - d(\pi; \mathcal{D}), \quad (38)$$

where Ω is computed using Equations 8, 10, and 32. Therefore, with a small batch of data, Equations 37 and 38 will choose a small k and as we acquire more data k naturally grows. To solve Equation 38 we follow Joseph et al. (2013) and use standard gradient decent with random restarts.

5 Empirical Results

Joseph et al. (2013) demonstrated the utility of a maximum return (MR) learner on a variety of simulated domains and a real-world problem with extremely complex dynamics. As the results from Joseph et al. (2013) show, MR performs poorly with little data and in this section we empirically demonstrate how Structural Return Maximization (SRM) remedies this shortcoming in a variety of domains.

The evaluation of SRM is done in comparison to MR due to MR being the only algorithm, to the best of our knowledge, in the RL literature which has identical assumptions to SRM. For each experiment, the chosen approaches⁸ choose policies from a designer-provided policy class. To use SRM we imposed a structure on the policy class (where the original policy class was the largest, most expressive class in the structure) and the methodology from Section 4.6 allowed SRM to select an appropriately sized policy class and policy from that class. The MR learner maximizes the empirical return (Equation 15) using the single, largest policy class.

We compare these approaches on three simulated problems: a 1D toy domain, the inverted pendulum domain, and an intruder monitoring domain. The domains demonstrate how SRM naturally grows the policy class as more data is seen and is far less vulnerable to over-fitting with small amounts of data than a MR learner. Policy classes were comprised of linear combinations of radial basis functions (Section 4.5) and training data was collected from a random policy. To piece together artificial trajectories for MFMC we used $\tilde{N} = (0.1)N$ where N is the number of data episodes.

5.1 1D Toy Domain

The purpose of the 1D toy domain is to enable understanding for the reader. The domain is a single dimensional world consisting of an agent who begins at $s_0 = 0$ and attempts to “stabilize” at $s = 0$ in the presence of noise. The dynamics are $s_{t+1} =$

⁸See Section 1 for our discussion on the pitfalls of using model-based approaches in this setting.

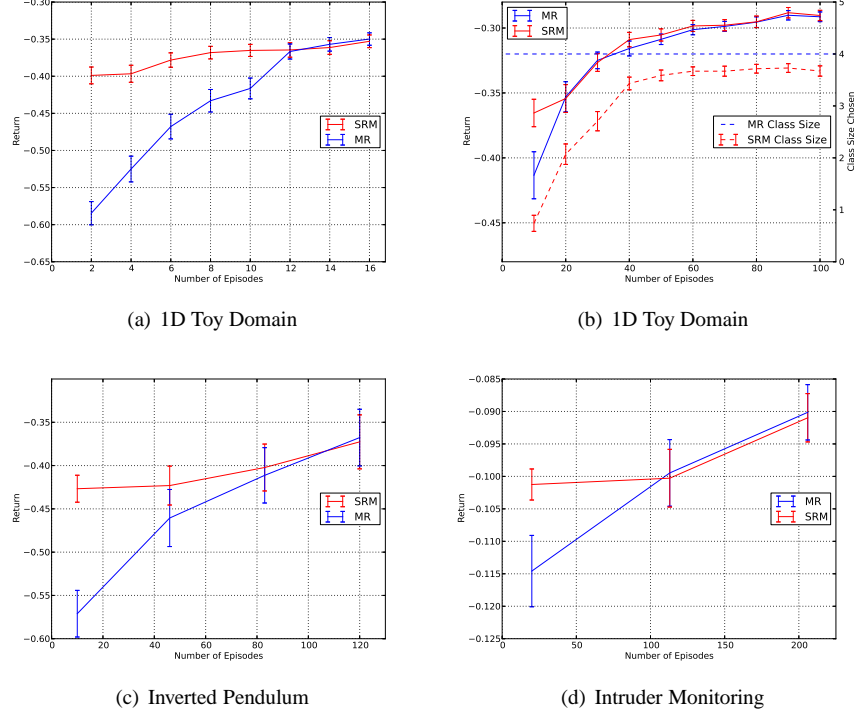


Figure 1: Performance versus the amount of training data and class size versus the amount of training data (a, b) on the 1D toy domain. Performance versus the amount of training data on the inverted pendulum domain (c) and the intruder monitoring domain (d). Error bars represent the 95% confidence interval of the mean.

$s_t + a + e$, $s_t \in [-1, 1]$ and e is a uniform random variable over $[-1/4, 1/4]$. The agent takes actions $a \in [-0.5, 0.5]$ and has reward function $\rho(s_t) = 5|s_t|$. For the policy representation we used four evenly spaced radial basis functions and for SRM we imposed five limits $l_k \in \{0, 0.125, 0.25, 0.375, 0.5\}$ on $|\phi_i|$ (see Equation 35).

Figures 1(a) and 1(b) show the performance of SRM (red solid line) and MR (blue solid line) on the 1D toy domain, where figure 1(a) is zoomed in to highlight the SRM's performance with small amounts of data. The plot shows that MR over-fits the small amounts of data, resulting in poor performance. On the other hand, SRM overcomes the problem of over-fitting by selecting a policy class which is appropriately sized for the amount of data. Figure 1(b) illustrates how SRM (red dashed line) selects a larger policy class as more data is seen, in contrast to MR, which uses a fixed policy class (blue dashed line). The figures show that as SRM is given more data, it selects increasing larger classes, allowing it learn higher performing policies without over-fitting.

5.2 Inverted Pendulum

The inverted pendulum is a standard RL benchmark problem (see Lagoudakis & Parr (2003b) for a detailed explanation and parameterization of the system). In our experiments we started the pendulum upright with the objective of learning policies which stabilize the pendulum in the presence of noise. For the policy representation we placed 16 evenly spaced radial basis functions and for SRM we imposed two limits on $|\phi_i|$ (Equation 35), $l_k \in \{0, 50\}$.

Figure 1(c) shows the performance of SRM (red) and MR (blue) on inverted pendulum. Similar to the results on the 1D toy domain, we see that MR over-fits the training data early on, resulting in poor performance. In contrast, SRM achieves higher performance early on by using a small policy class with small amounts of data and growing the policy class as more data is seen.

5.3 Intruder Monitoring

The intruder monitoring domain models the scenario of an intruder transversing a two dimensional world where a camera must monitor the intruder around a sensitive location. The camera observes a circle of radius $r_{\text{cam}} = 0.5$ centered at s_{cam} , and the intruder, located at s_I , wanders toward the sensitive location with additive uniform noise. The camera dynamics follow $s_{\text{cam}} = s_{\text{cam}} + a$, where the agent takes action $a \in [-0.1, 0.1]$ and has reward $\rho(s_{\text{cam}}, s_I) = \sum_i \frac{\min(\|s_{\text{cam}} - X_i\|, 0.5)}{\max(\|s_{\text{sensitive}} - X_i\|, 0.05)}$ where $s_{\text{sensitive}} = [0, 0]$. For our policy representation, we placed 16 radial basis points on a grid inside $[-1, 1], [-1, 1]$ and for SRM we imposed two limits on $|\phi_i|$, $l_k \in \{0, 0.1\}$.

Figure 1(d) shows the performance of SRM (red) and MR (blue) on inverted pendulum. Similar to the results on the previous domains, we see that SRM outperforms MR due to using a small policy class with small amounts of data and growing the policy class as more data is seen.

6 Related Work

While there has been a significant amount of prior work relating Reinforcement Learning (RL) and classification (Langford & Zadrozny, 2003; Lagoudakis & Parr, 2003a; Barto & Dietterich, 2004; Langford, 2005), to the best of our knowledge, our work is the first to sufficiently develop the mapping to allow the analysis presented in Section 2 to be transferred to RL. There has also been work to use classifiers to represent policies in RL (Bagnell et al., 2003; Rexakis & Lagoudakis, 2008; Dimitrakakis & Lagoudakis, 2008; Blatt & Hero, 2006), which is tangential to our work; our focus is on using the principle Structural Risk Minimization for RL. Additional work uses classification theory to bound performance for on-policy data (Lazaric et al., 2010; Farahmand et al., 2012), for which Section 3.1.3 can be seen as extending to batch, off-policy data.

A second class of approaches aim to prevent over-fitting with small amounts of data by using both frequentist and Bayesian forms of regularization Strens (2000); Abbeel & Ng (2005); Bartlett & Tewari (2009); Doshi-Velez et al. (2010). These methods either lack formal guarantees similar to Equation 29 for batch data or require strong

assumptions about the form of the true dynamics model, the true value function, or the optimal policy.

The RL literature also has a great deal of work growing representations as more data is seen. Past work in the model-based (Doshi-Velez, 2009; Joseph et al., 2011) and value-based (Ratitch & Precup, 2004; Whiteson et al.; Geramifard et al., 2011) settings have proven successful but generally require either prior distributions or a large amount of training data collected under a specific policy. Additionally, our approach applies in model-based, value-based, and policy search settings by treating either the model class or value function class as indirect policy representations.

7 Conclusion

In this work we applied Structural Risk Minimization to Reinforcement Learning (RL) to allow us to learn appropriately sized policy classes for a given amount of batch data. The resulting algorithm had provable performance bounds under extremely weak assumptions. To accomplish this we presented a mapping of classification to RL which allowed us to transfer the theoretical bounds previously developed in the context of classification. These bounds allowed us to learn the policy from a structured policy class which maximized the bound on return. We demonstrated the benefit of our approach on a 1D toy, inverted pendulum, and intruder monitoring domains as compared to an agent which naively maximizes the empirical return of the single, large policy class.

References

- Abbeel, Pieter and Ng, Andrew Y. Exploration and apprenticeship learning in reinforcement learning. In *ICML*, 2005.
- Anthony, Martin and Bartlett, Peter L. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.
- Asian, Ozlem, Yildiz, Olcay Taner, and Alpaydin, Ethem. Calculating the vc-dimension of decision trees. In *ISCIS 2009, 14-16 September 2009, North Cyprus*, pp. 193–198. IEEE, 2009.
- Bagnell, J. Andrew, Kakade, Sham, Ng, Andrew, and Schneider, Jeff. Policy search by dynamic programming. In *NIPS*, 2003.
- Bartlett, Peter L. and Mendelson, Shahar. Rademacher and gaussian complexities: risk bounds and structural results. *Journal of Machine Learning*, 2003.
- Bartlett, Peter L. and Tewari, Ambuj. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. In *UAI*, 2009.
- Bartlett, Peter L., Boucheron, Stéphane, and Lugosi, Gábor. Model selection and error estimation. *Machine Learning*, 48(1-3):85–113, 2002a.

- Bartlett, Peter L., Bousquet, Olivier, and Mendelson, Shahar. Local rademacher complexities. In *Annals of Statistics*, pp. 44–58, 2002b.
- Barto, A.G. and Dietterich, T.G. Reinforcement learning and its relationship to supervised learning. *Handbook of learning and approximate dynamic programming*, 2004.
- Baxter, Jonathan and Bartlett, Peter L. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 2001.
- Bertsekas, Dimitri P. *Dynamic Programming and Optimal Control*. Athena Scientific, 2000.
- Blatt, Doron and Hero, Alfred. From weighted classification to policy search. In Weiss, Y., Schölkopf, B., and Platt, J. (eds.), *NIPS*. 2006.
- Cherkassky, Vladimir S. and Mulier, Filip. *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1998.
- Dimitrakakis, Christos and Lagoudakis, Michail G. Rollout Sampling Approximate Policy Iteration. *Machine Learning*, 72(3):157–171, September 2008.
- Doshi-Velez, Finale. The infinite partially observable markov decision process. In *NIPS*, 2009.
- Doshi-Velez, Finale, Wingate, David, Roy, Nicholas, and Tenenbaum, Joshua. Non-parametric bayesian policy priors for reinforcement learning. 2010.
- Farahmand, Amir-Massoud, Precup, Doina, and Ghavamzadeh, Mohammad. Generalized classification-based approximate policy iteration. In *European Workshop on Reinforcement Learning*, 2012.
- Fonteneau, Raphael, Murphy, Susan A., Wehenkel, Louis, and Ernst, Damien. Model-free monte carlo-like policy evaluation. *Journal of Machine Learning Research - Proceedings Track*, 2010.
- Fonteneau, Raphael, Murphy, SusanA, Wehenkel, Louis, and Ernst, Damien. Batch mode reinforcement learning based on the synthesis of artificial trajectories. 2012.
- Geramifard, Alborz, Doshi, Finale, Redding, Joshua, Roy, Nicholas, and How, Jonathan. Online discovery of feature dependencies. In *ICML*, 2011.
- Hoeffding, Wassily. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
- Joseph, Joshua, Doshi-Velez, Finale, Huang, Albert S., and Roy, Nicholas. A Bayesian Nonparametric Approach to Modeling Motion Patterns. *Autonomous Robots*, 31(4): 383–400, 2011.
- Joseph, Joshua, Geramifard, Alborz, Roberts, John W., How, Jonathan P., and Roy, Nicholas. Reinforcement learning with misspecified model classes. In *ICRA*, 2013.

- Lagoudakis, Michail G. and Parr, Ronald. Reinforcement learning as classification: Leveraging modern classifiers. In *ICML*, 2003a.
- Lagoudakis, Michail G. and Parr, Ronald. Least-squares policy iteration. *JMLR*, 4: 1107–1149, 2003b.
- Langford, John. Relating reinforcement learning performance to classification performance. In *ICML*, 2005.
- Langford, John and Zadrozny, Bianca. Reducing t-step reinforcement learning to classification, 2003.
- Lazaric, A., Ghavamzadeh, M., Munos, R., et al. Analysis of a classification-based policy iteration algorithm. 2010.
- Lee, Wee Sun, Bartlett, Peter, and Williamson, Robert. Lower bounds on the vc-dimension of smoothly parametrized function classes. *Neural Computation*, 1995.
- McDonald, Daniel, Shalizi, Cosma, and Schervish, Mark. Estimated vc dimension for risk bounds. 2011.
- Menache, Ishai, Mannor, Shie, and Shimkin, Nahum. Basis function adaptation in temporal difference reinforcement learning. *Annals OR*, 134(1):215–238, 2005.
- Meuleau, Nicolas, Peshkin, Leonid, Kaelbling, Leslie, and Kim, Kee. Off-policy policy search. Technical report, MIT Artificial Intelligence Laboratory, 2000.
- Ratitch, Bohdana and Precup, Doina. Sparse distributed memories for on-line value-based reinforcement learning. In *Machine Learning: ECML*, Lecture Notes in Computer Science, 2004.
- Rexakis, Ioannis and Lagoudakis, Michail G. Classifier-based policy representation. In *ICMLA*, 2008.
- Shao, Xuhui, Cherkassky, Vladimir, and Li, William. Measuring the vc-dimension using optimized experimental design. *Neural Computation*, 12:2000, 1999.
- Shawe-Taylor, John, Holloway, Royal, Bartlett, Peter L., Williamson, Robert C., and Anthony, Martin. Structural risk minimization over data-dependent hierarchies, 1996.
- Strens, Malcolm. A bayesian framework for reinforcement learning. In *ICML*, pp. 943–950, 2000.
- Sutton, Richard S. and Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, May 1998.
- Vapnik, Vladimir. *Statistical learning theory*. Wiley, 1998. ISBN 978-0-471-03003-4.
- Vapnik, Vladimir, Levin, Esther, and LeCun, Yann. Measuring the vc-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994.

Vapnik, Vladimir N. *The nature of statistical learning theory*. Springer-Verlag New York, Inc, 1995.

Whiteson, Shimon, Taylor, Matthew, and Stone, Peter. Adaptive tile coding for value function approximation. Technical report, University of Texas at Austin.